Raspberry Pi Setup

Change History

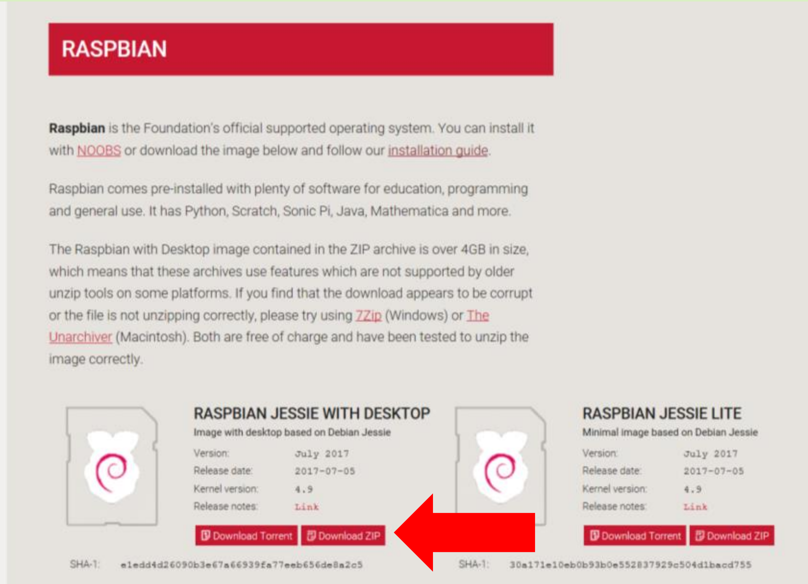
Version	Author	Date	Description
0.1	Adrian Gfeller	22.05.2017	Initial Draft
1.0	Walter Rothlin	20.07.2017	Updated
1.1	Benjamin Raison	22.09.2017	Fixed erros, formatting, replaced apt-get with apt where possible, added "Installing Apache"
1.2	Walter Rothlin	23.09.2017	Updated example to python 3
1.3	Benjamin Raison	28.09.2017	Added Perl CGI + example, added instructions on sensor input errors
1.4	Benjamin Raison	09.10.2017	Added section on discovering the IP address & Pi Plates
1.5	Walter Rothlin		Set-UP WiFi an BSU
1.6	Walter Rothlin	29.11.2017	Web-Server Config
1.7	Walter Rothlin	14.06.2019	CGI Config
1.8	Walter Rothlin	06.03.2021	Added Remote-Access

Contents

Download OS for Raspberry

In this guide, *Raspbian* will be used which is a Linux distribution based on Debian and targeted at Raspberry Pi. Raspbian comes in to versions: DESKTOP and LITE.

Download the latest *Raspbian Jessie Lite* from https://www.raspberrypi.org/downloads/raspbian/ (20.7.2017)



RASPBIAN

Raspbian is the Foundation's official supported operating system. You can install it with NOOBS or download the image below and follow our installation guide.

Raspbian comes pre-installed with plenty of software for education, programming and general use. It has Python, Scratch, Sonic Pi, Java, Mathematica and more.

The Raspbian with Desktop image contained in the ZIP archive is over 4GB in size, which means that these archives use features which are not supported by older unzip tools on some platforms. If you find that the download appears to be corrupt or the file is not unzipping correctly, please try using 7Zip (Windows) or The Unarchiver (Macintosh). Both are free of charge and have been tested to unzip the image correctly.

RASPBIAN JESSIE WITH DESKTOP
Image with desktop based on Debian Jessie
Version: July 2017
Release date: 2017-07-05
Kernel version: 4.9
Release notes: Link

SHA-1: e1edd4d26090b3e67a66939fa77eeb656de8a2c5

RASPBIAN JESSIE LITE
Minimal image based on Debian Jessie
Version: July 2017
Release date: 2017-07-05
Kernel version: 4.9
Release notes: Link

SHA-1: 30a171e10eb0b93b0e552837929c504d1bacd755

After the download has finished, extract the zip folder to get to the image file.

Write Image to SD-Card

The following information is extracted from https://www.raspberrypi.org/documentation/installation/installing-images/README.md and https://www.raspberrypi.org/documentation/installation/installing-images/README.md which is linked on the download page.

Install Win32DiskImager from https://sourceforge.net/projects/win32diskimager/ (20.07.2017)

Insert the SD Card into the PC (Via integrated card reader or an adapter)

If you have to format it use FAT32.

Run Win32DiskImager.exe and finish the installation.

Run the program and set the Image to the previously extracted .img file and select the drive letter. Be careful with selecting this letter as you can lose data when you're choosing the wrong one.

Then click on Write

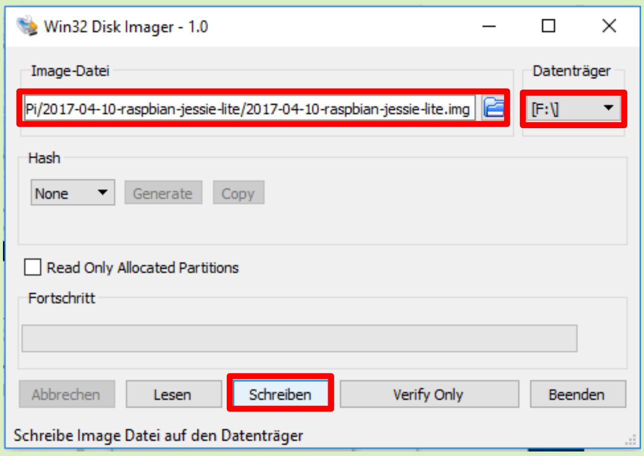


Image-Datei: Pi/2017-04-10-raspbian-jessie-lite/2017-04-10-raspbian-jessie-lite.img

Datenträger: [F:\]

Hash: None

□ Read Only Allocated Partitions

Schreibe Image Datei auf den Datenträger

The SD Card is now ready to be inserted into the Raspberry Pi

First startup

Boot & Login

Insert the SD-Card into the Raspberry Pi, plug in an HDMI Cable and provide it with power over a MicroUSB cable.

You'll be logged in automatically and will be presented with the desktop.
Don't connect to internet yet!!! Your Pi is totally unsecured at the moment.
The first and most important step is to change at least the password because the whole world knows it at this moment and as soon as you're connected to the internet, everyone is able to get access to your Raspberry Pi.
But before that step, we need to make sure that our keyboard layout is matching our keyboard. Otherwise you'll enter your password wrong and can't log in when the layout changes.

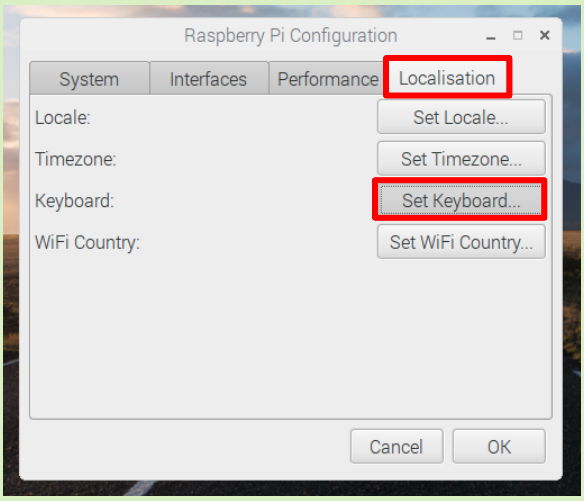
In case you're not automatically logged in:

raspberrypi login (username): **pi**
Password: **raspberry** (type "z" instead of "y" because of UK keyboard layout)

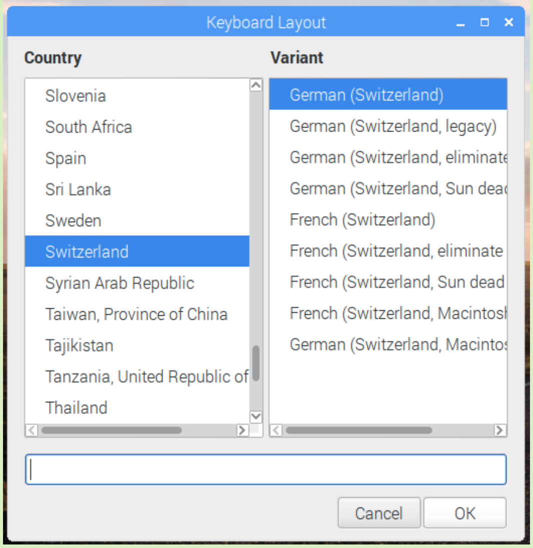
Changing the Keyboard Layout

Run raspi-config by opening the *Start Menu* and then *Preferences* > *Raspberry Pi Configuration*

Navigate to the *Localisation* Tab and click on *Set Keyboard…*

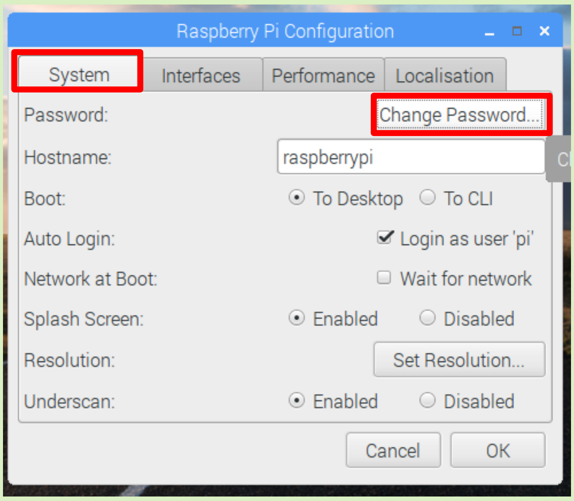


Another Window should pop-up. Select *Switzerland* and *German (Switzerland)* / whatever is matching the keyboard layout. Then click *OK*. After this the layout has already changed.


Keyboard Layout
Country
Slovenia
South Africa
Spain
Sri Lanka
Sweden
Switzerland
Syrian Arab Republic
Taiwan, Province of China
Tajikistan
Tanzania, United Republic of
Thailand
Variant
German (Switzerland)
German (Switzerland, legacy)
German (Switzerland, eliminate
German (Switzerland, Sun dead
French (Switzerland)
French (Switzerland, eliminate
French (Switzerland, Sun dead
French (Switzerland, Macintosh
German (Switzerland, Macintos
Cancel
OK

Changing the Password

Again, in the Raspberry Pi Configuration, activate the *System* Tab and click on the *Change Passwor…* button:



Raspberry Pi Configuration

System | Interfaces | Performance | Localisation

Password: Change Password...

Hostname: raspberrypi

Boot: ● To Desktop ○ To CLI

Auto Login: ☑ Login as user 'pi'

Network at Boot: □ Wait for network

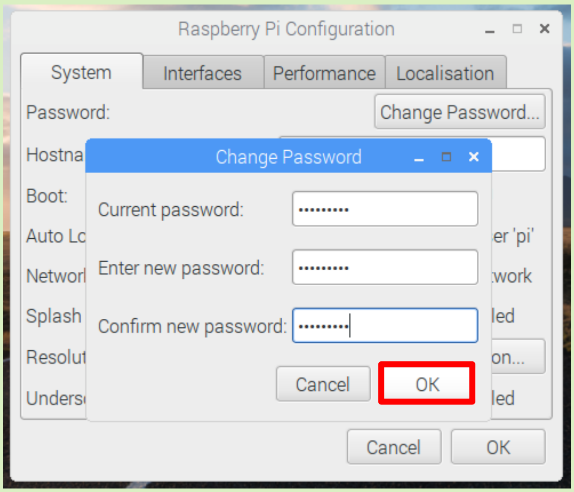
Splash Screen: ● Enabled ○ Disabled

Resolution: Set Resolution...

Underscan: ● Enabled ○ Disabled

Cancel OK

As *Current password* enter **raspberry** then assign a new one; two times to confirm. Then click on *OK*.



Change Password

Current password: •••••••••

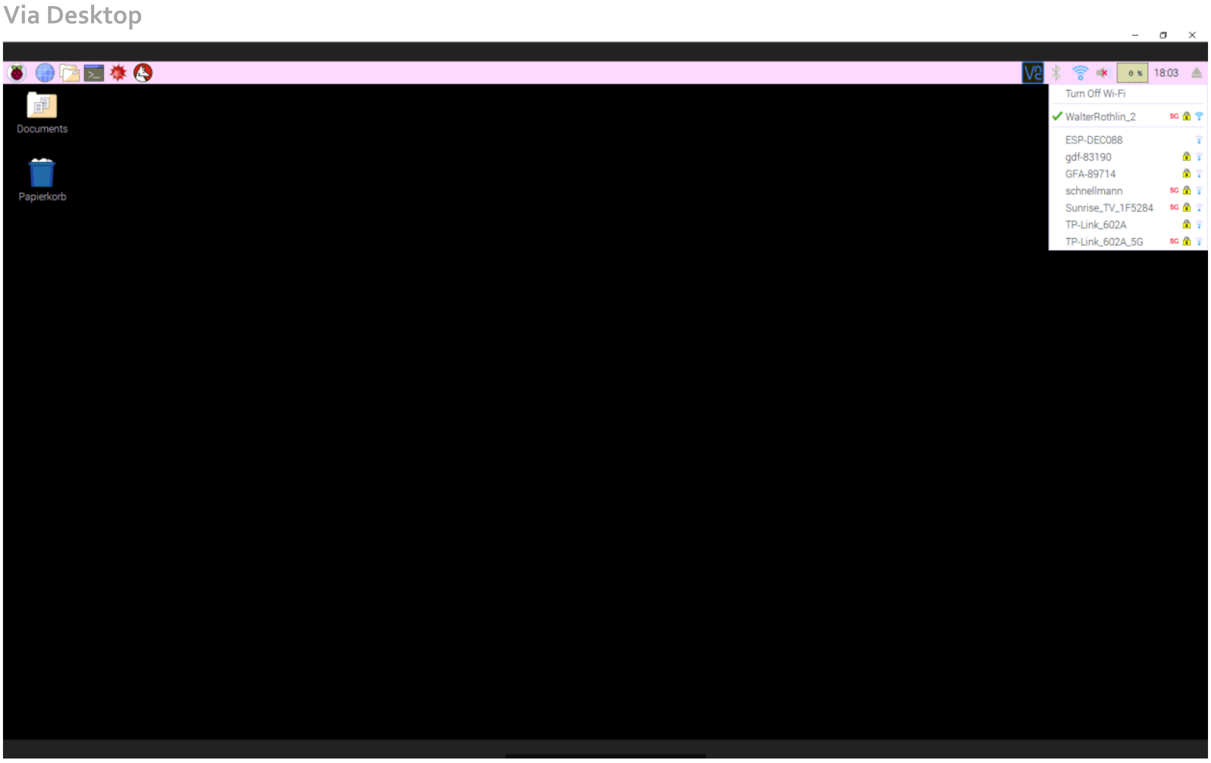
Enter new password: •••••••••

Confirm new password: •••••••••

Cancel OK

Connecting to internet

Via Desktop

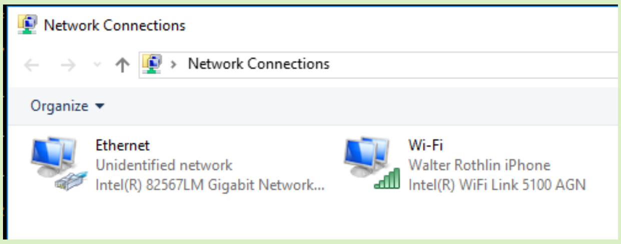


Via cable

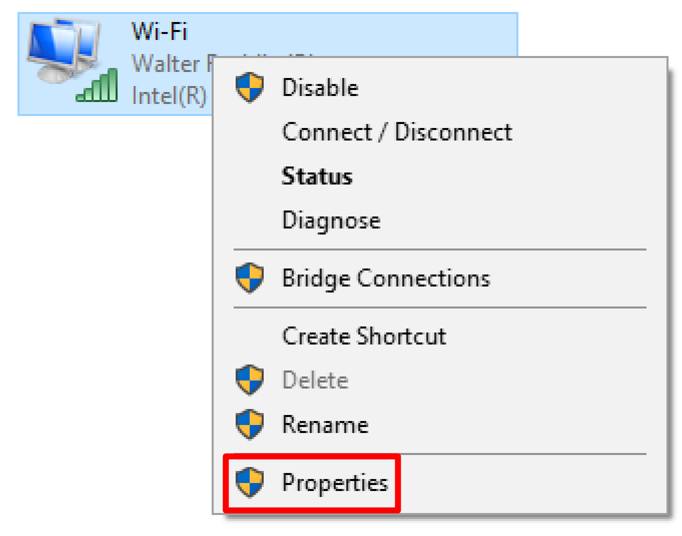
This is easy: Just plug the cable into the Pi and the router will assign an IP address to the Pi.

Should you be unable to access the internet, do the following (on Windows):

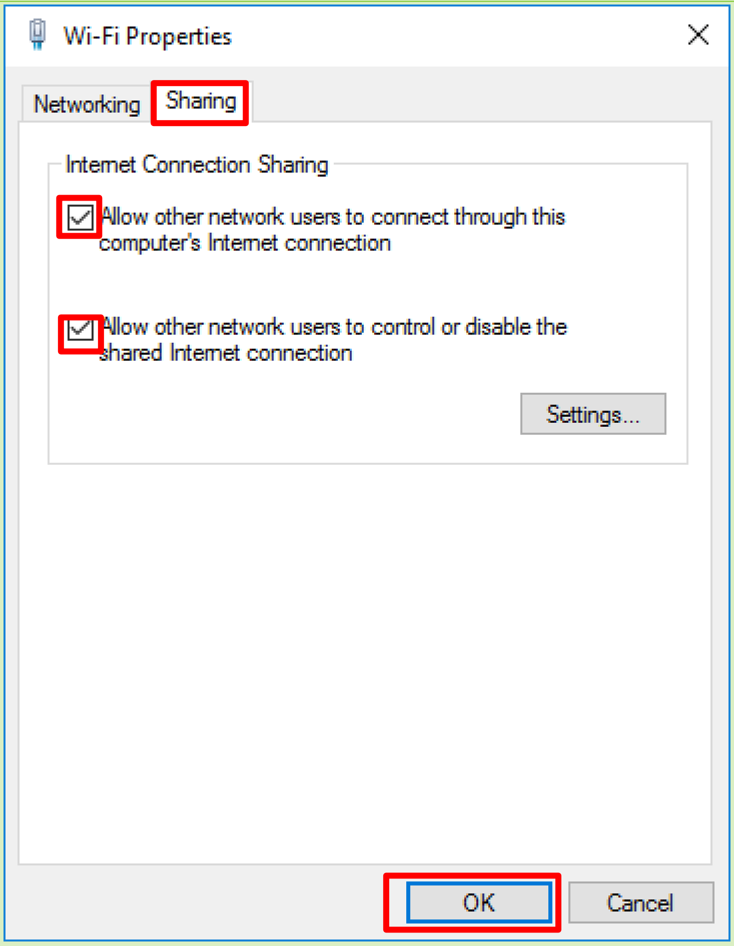
Navigate to the control panel and open "Network Connections"



Select the network you already have an internet connection on and open the properties.



Then, open the "Sharing" tab and check both boxes, then OK.



Wi-Fi Properties

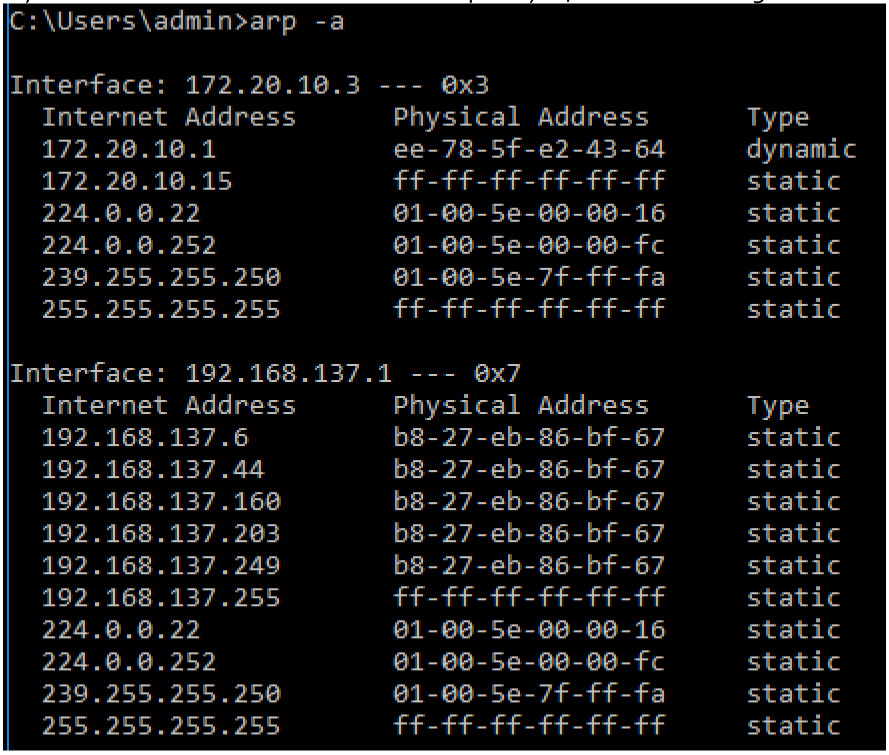
Internet Connection Sharing

☑ Allow other network users to connect through this computer's Internet connection

☑ Allow other network users to control or disable the shared Internet connection

Then restart the Raspberry Pi.

If you don’t know the IP address of the Raspberry Pi, run the following command on the host computer:



```
C:\Users\admin>arp -a

Interface: 172.20.10.3 --- 0x3
  Internet Address      Physical Address      Type
  172.20.10.1           ee-78-5f-e2-43-64     dynamic
  172.20.10.15          ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.137.1 --- 0x7
  Internet Address      Physical Address      Type
  192.168.137.6         b8-27-eb-86-bf-67     static
  192.168.137.44        b8-27-eb-86-bf-67     static
  192.168.137.160       b8-27-eb-86-bf-67     static
  192.168.137.203       b8-27-eb-86-bf-67     static
  192.168.137.249       b8-27-eb-86-bf-67     static
  192.168.137.255       ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Then try all IP addresses where the “Physical Address” starts with **b8-27-eb**. On of them will be the Rasperry Pi.

Via Wi-Fi (via Command line)

Raspberry 3 Model B has on board WIFI otherwise plug in a WIFI dongle into a USB port of the pi.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
network={
    ssid="Walter Rothlin iPhone"
    psk="testingPassword"
}
```

Detail description under:
https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md

```
sudo iwlist wlan0 scan      # this command will list all available networks
```

Via BZU Wi-Fi

1. In der datei : `/etc/wpa_supplicant/wpa_supplicant.conf` folgendes einfügen (benutzername und passwort auf ihres abändern!)

   a. Mit `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`

2.
```
network={
            ssid="BZU"
            proto=RSN
            key_mgmt=WPA-EAP
            pairwise=CCMP
            auth_alg=OPEN
            eap=PEAP
            identity="vorname.nachname"
            password="ihrbzupasswort"
            phase2="auth=MSCHAPV2"
    }
```

3. Dann führen sie noch folgende 3 Befehle aus:

   a. root@raspberrypi:~# `wpa_cli reconfigure` // lädt die Konfigurationsdatei neu

   b. root@raspberrypi:~# sudo `ifconfig wlan0 up` // setzt das wlano device auf aktiv

   c. root@raspberrypi:~# `ifconfig wlan0` // hier sollten sie nun eine IP Adresse sehen

Damit das ganze automatisch funktioniert, ändern sie die folgende Zeile in `/etc/network/interfaces:`

```
iface wlan0 inet manual
```
auf:
```
iface wlan0 inet dhcp
```

Wi-Fi set-up using desktop

Raspberry 3 Model B has on board WIFI otherwise plug in a WIFI dongle into a USB port of the pi.

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
network={
    ssid="Walter Rothlin iPhone"
    psk="testingPassword"
}
```

Detail description under:
https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md

```
sudo iwlist wlan0 scan     # this command will list all available networks
```

Updating the System

First thing after connecting to the internet, is to update all the software and performing possible kernel-updates. This will ensure, that known bugs or security holes are fixed.
This is done with these two apt-get commands:

```
sudo apt update    # list all the modules which will be upgraded (doesn’t perform an change on the system)

sudo apt upgrade
```

The `sudo apt autoremove` command can be issued to remove unnecessary package files.

Enable On-Board-Screen

Information extracted from the **Tontec MZ61581** setup guide.

Open /boot/config.txt

```
sudo nano /boot/config.txt
```

And add these lines to the bottom

```
dtparam=spi=on
dtoverlay=mz61581
```

Then save and reboot
Here we will change the default output display from HDMI to Tontec Screen

```
sudo nano /usr/share/X11/xorg.conf.d/99-fbturbo.conf
```

Change
    Option   "fbdev" "/dev/fb0"
To

```
        Option          "fbdev"  "/dev/fb1"
```

If you want to switch back to the HDMI display, just change it back to **fb0**

Enable SSH

Open up the Raspberry Pi Config again and navigate to the *Interfaces* Tab.

There activate the SSH service.

Save the Image of the Raspberry Pi

Create a new file with an *.img* extension somewhere on your PC.
Eject the SD-Card from you Raspberry Pi and plug it into your computer.

Open up Win32DiskImager and for the image file select the empty one you've created before. Select the correct drive letter where the SD-Card is mounted and then click on *Read / Lesen*.

Remote-Access
Sobald der Raspberry am Netz hängt (via WiFi oder LAN) kann Remote darauf zugegriffen werden. Dazu ist wichtig, die IP Adressen zu kennen.

IP-Adresse Raspberry: ifconfig
IP-Adresse Desktop: ipconfig

a) Remote-Shell

Eine ssh Session vom Desktop zum Raspberry via PuTTY kreieren.
Damit können Commands abgesetzt und ASCII basierte Resultate angezeigt werden
Mit fire-ftp (Firefox plugin) oder Notepad++ (NPP FTP Plugin) können Files verschoben werden
Es ist wichtig je nach File-Type den Transfer-Mode (BIN/ASCII) zu beachten!

b) X11 Redirect (X11 ist das Windowing System auf Linux / UNIX)

X-Server:Desktop / Windows-Computer

X-Client: Remote-System, in unserem Fall der RaspberryPi

Auf dem Desktop ein X-Server starten (z.B. XMing), Es wird nichts sichtbar!
Eine ssh Session vom Desktop zum Raspberry via PuTTY kreieren
In der ssh Session folgendes auf dem Raspberry vorbereiten:

```
sudo raspi-config
```

Started config menu in Ascii Mode

Menu 5 / P3 VNC   Enable X forwarding

```
xhost+
```

Erlaubt, dass X11-Windows auf einem anderen Rechner (X-Server) angezeigt werden können

----

```
export DISPLAY=192.199.18.1
```

X11-Windows werden auf 192.199.18.1 (X-Server) umgeleitet / geöffnet

----

```
lxsession &
```

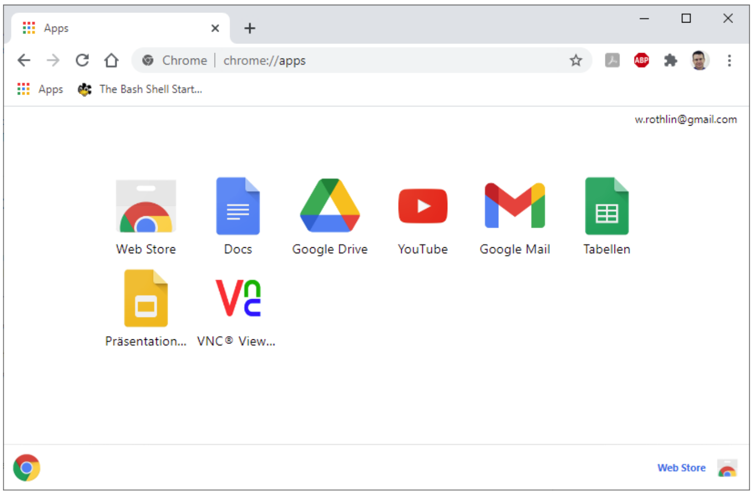
Starten der Windows-Oberfläche remote (Display umgeleitet via IP)

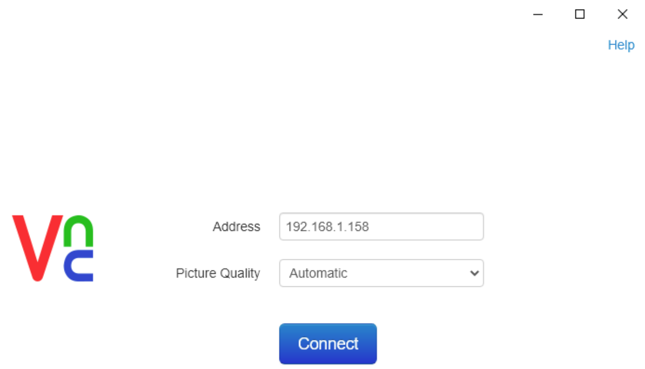
Sobald Sie Remote auf den RaspberryPi zugreifen können, können Sie z.B. Notepad++ mit dem NPP-FTP Plugin verwenden und dann die Files direkt auf dem RaspberryPi von ihrem PC aus editieren.

In Python gibt das Einrücken die Struktur des Programms vor. Ich empfehle, im Notepad++ unter Setting --> Preferences… -> Language resp. TAB Settings dies entsprechend einzustellen, so dass z.B. bei der Eingabe eines TAB 4 Spaces eingeführt werden.

Google VNC

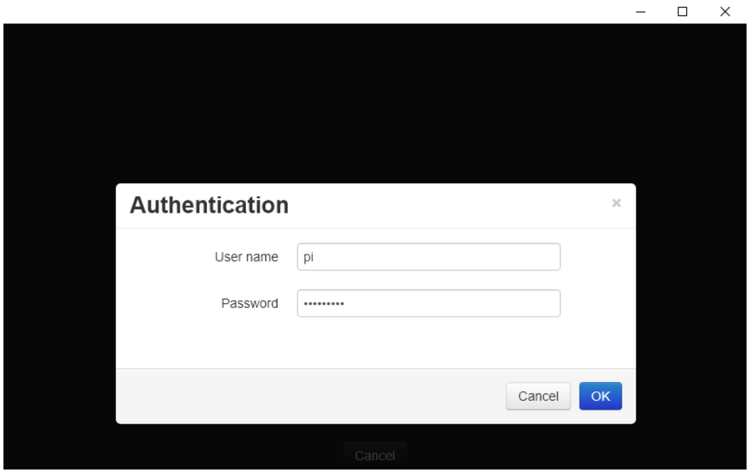
Run google-Chrome and start VNC





Address 192.168.1.158

Picture Quality Automatic



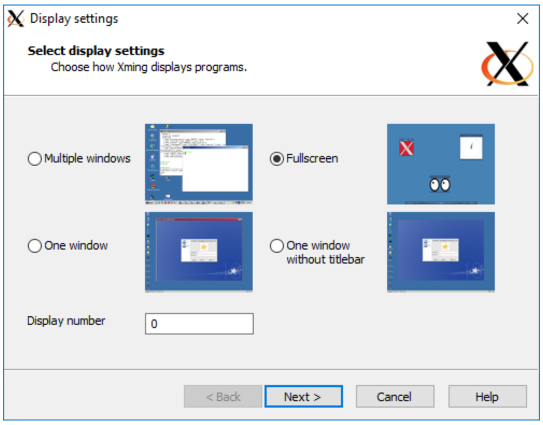
Authentication

User name pi

Password •••••••••

Remote Desktop with Xming and Putty

1. Download and Install Xming https://sourceforge.net/projects/xming/
2. Execute XLaunch
3. Choose "Fullscreen"


Display settings

Select display settings

Choose how Xming displays programs.

○ Multiple windows

● Fullscreen

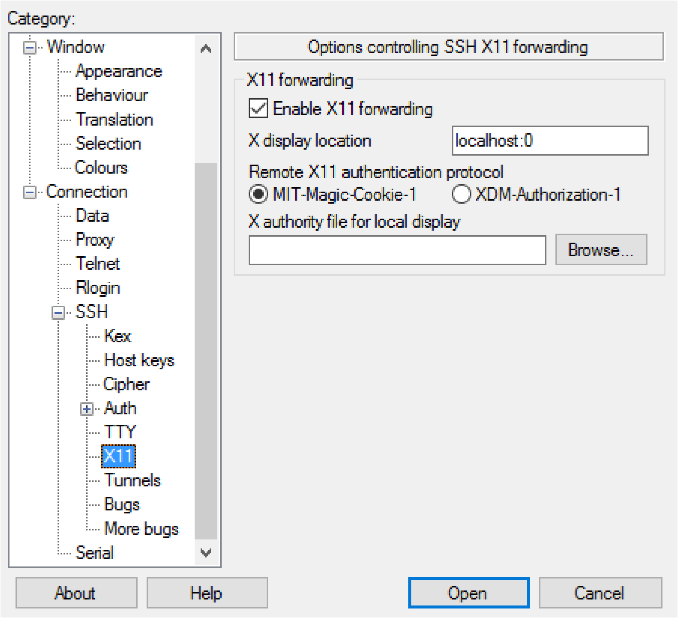
○ One window

○ One window without titlebar

Display number: 0

4. Press Enter multiple times until the window closes
   A new window should open in fullscreen

5. Tab out and go to Putty
6. Load your connection (not open!)
7. Go to **Connection -> SSH -> X11**, enable X11 forwarding and set X display location to "localhost:0"


Options controlling SSH X11 forwarding

X11 forwarding

☑ Enable X11 forwarding

X display location: localhost:0

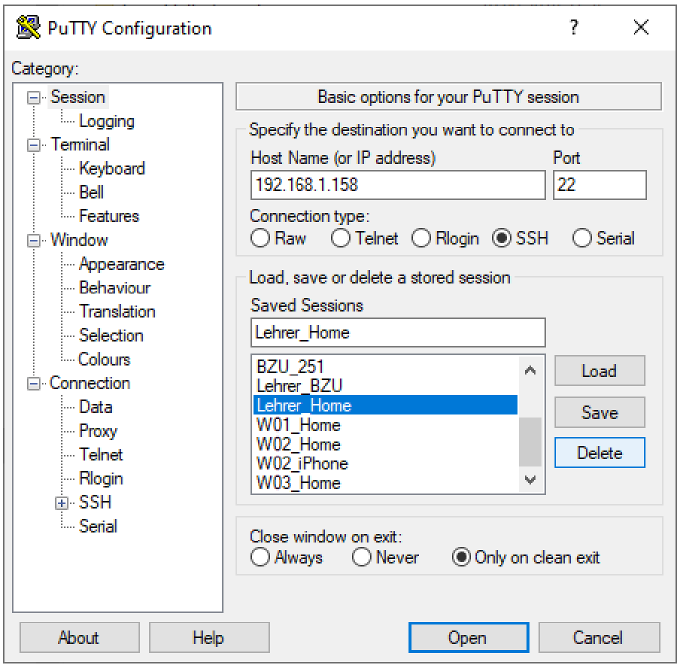
Remote X11 authentication protocol

● MIT-Magic-Cookie-1 ○ XDM-Authorization-1

X authority file for local display

8. Save your connection
9. Open your connection
10. In the terminal, type "**startlxde**" and press Enter
11. After a short while, it should switch to Xming and start displaying the desktop.

ssh using PuTTY



Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.158

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

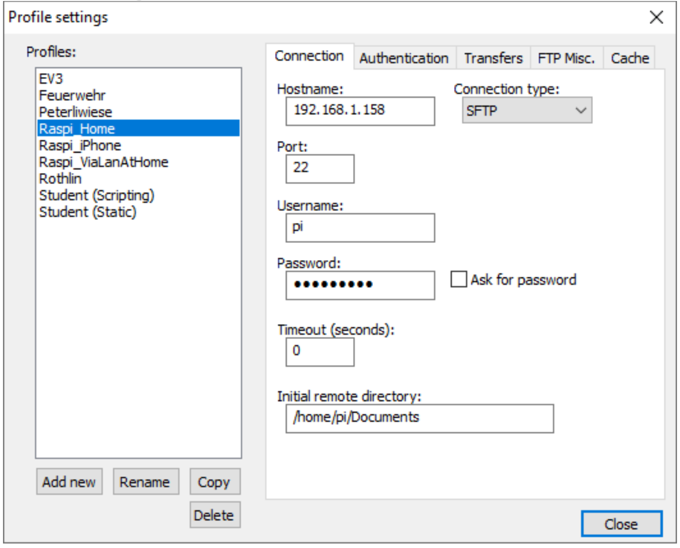
Load, save or delete a stored session

Saved Sessions: Lehrer_Home

BZU_251
Lehrer_BZU
Lehrer_Home
W01_Home
W02_Home
W02_iPhone
W03_Home

Close window on exit: ○ Always ○ Never ● Only on clean exit

nppFTP plugin for Notepad++



Profiles:

EV3
Feuerwehr
Peterliwiese
Raspi_Home
Raspi_iPhone
Raspi_ViaLanAtHome
Rothlin
Student (Scripting)
Student (Static)

Hostname: 192.168.1.158

Connection type: SFTP

Port: 22

Username: pi

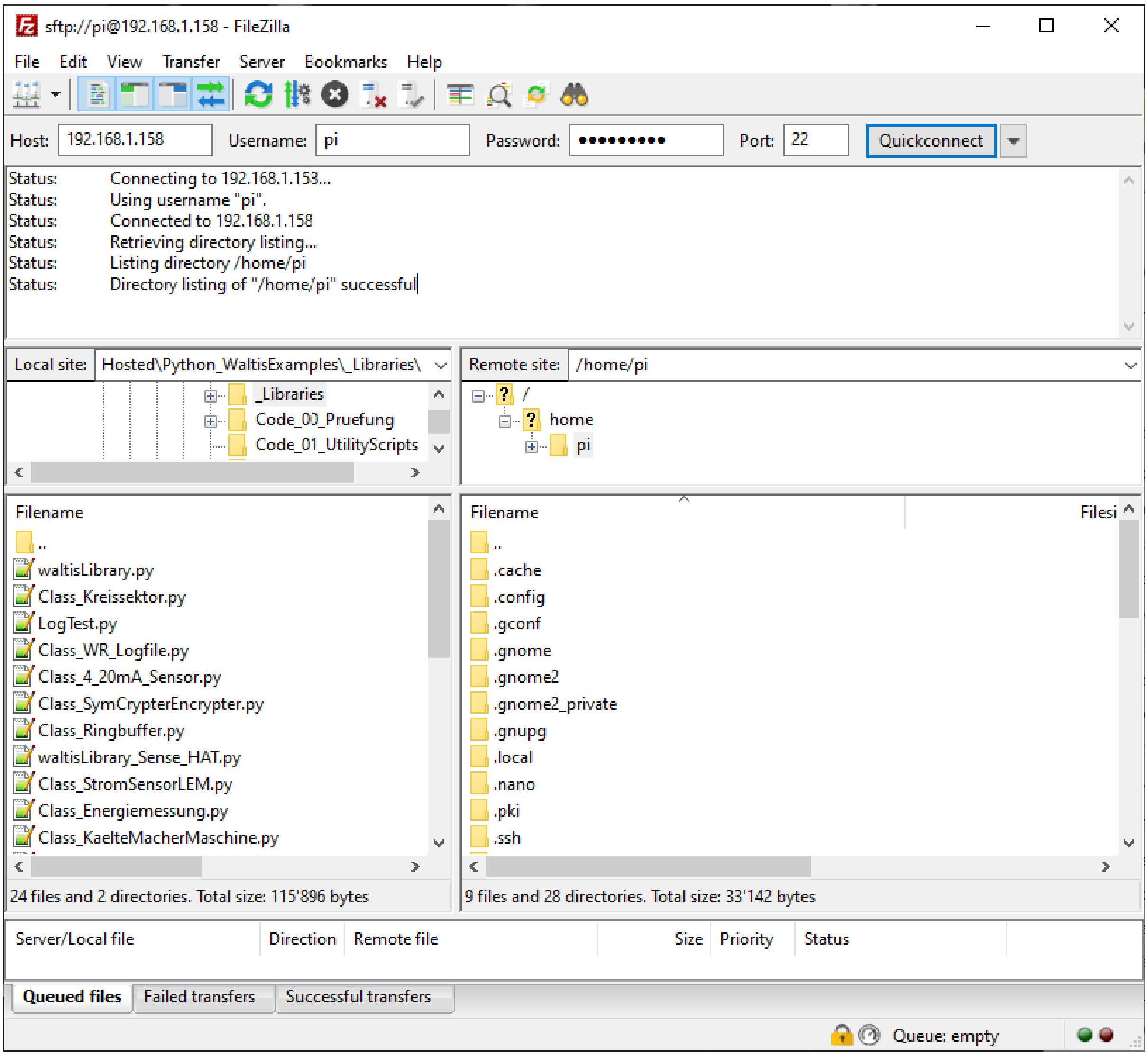
Password: •••••••••

□ Ask for password

Timeout (seconds): 0

Initial remote directory: /home/pi/Documents

SFTP via FileZilla



sftp://pi@192.168.1.158 - FileZilla
File Edit View Transfer Server Bookmarks Help
Quickconnect

Host: 192.168.1.158 Username: pi Password: ●●●●●●●●● Port: 22

```
Status:      Connecting to 192.168.1.158...
Status:      Using username "pi".
Status:      Connected to 192.168.1.158
Status:      Retrieving directory listing...
Status:      Listing directory /home/pi
Status:      Directory listing of "/home/pi" successful
```

Local site: Hosted\Python_WaltisExamples_Libraries\

Remote site: /home/pi

_Libraries
Code_00_Pruefung
Code_01_UtilityScripts
/
home
pi

Filename
..
waltisLibrary.py
Class_Kreissektor.py
LogTest.py
Class_WR_Logfile.py
Class_4_20mA_Sensor.py
Class_SymCrypterEncrypter.py
Class_Ringbuffer.py
waltisLibrary_Sense_HAT.py
Class_StromSensorLEM.py
Class_Energiemessung.py
Class_KaelteMacherMaschine.py

Filename
..
.cache
.config
.gconf
.gnome
.gnome2
.gnome2_private
.gnupg
.local
.nano
.pki
.ssh

24 files and 2 directories. Total size: 115'896 bytes

9 files and 28 directories. Total size: 33'142 bytes

Pi Plates

Run **sudo raspi-config** and enable **SPI** under **Advanced**

Then reboot and run one of the following commands:

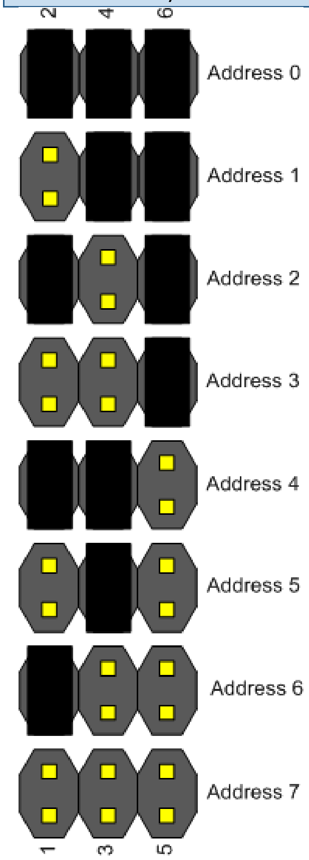
```
sudo pip install pi-plates  #Python < 3
sudo pip3 install pi-plates #Python >= 3
```

You can now use your Pi Plates.
Should your program complain about missing libraries, install them with **pip(3)** or **apt.**

Example programs can be found in ~/Documents/PythonExamples/Code_06…

Addressing the boards

Each plate in the stack must have a unique address. The address is set as seen in the image below. The address is only read at boot time, so you have to restart the Raspberry Pi to change the address of a plate.



Further documentation

For complete documentations, refer to http://pi-plates.com/documentation

Installing Apache

This will guide you in setting up a running webserver able to show both HTML pages and run Python scripts

Run the following command to install the webserver:

```
sudo apt install apache2
```

Enter the IP of your Raspberry Pi into a web browser to see if it works.

Enable CGI

Create a cgi-directory:

```
sudo mkdir /var/www/cgi-bin/
```

Create file /var/www/cgi-bin/test.py with the following content:

```
#!/usr/bin/python3

print(“Content-Type: text/html”)
print()
print(“Hello World!“)
```

The empty line is important. It separates the headers from the body.

Change apache config:

```
sudo nano /etc/apache2/apache2.conf
```

Add the following at the end of the config-file:

```
AddHandler cgi-script .cgi .py .pl
```

Then, look for something like this:

```
<Directory /var/www>
    Options Indexes FollowSymlinks
```

and add the following block:

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory /var/www/cgi-bin>
        Options Indexes FollowSymLinks ExecCGI
        AllowOverride None
        Require all granted
</Directory>
```

Save and exit the file. Then, run the following commands:

```
sudo a2enmod cgi
sudo service apache2 restart
```

Open `[you-ip-here]/cgi-bin/test.py` in your web browser.

Check log-file for Errors:

```
sudo tail -f /var/log/apache2/error.log
```

Enable Sense-Hat and PiPlates for CGI calls

Execute the following commands as **root**:

```
# Add apache to input group
sudo adduser www-data input
sudo adduser www-data kmem
sudo adduser www-data spi
sudo adduser www-data i2c
sudo adduser www-data gpio
```

If the sense HAT gives a “permission denied” error, run the following command to copy the sense hat config file to the apache home directory:

```
sudo mkdir /var/www/.config
sudo cp –r /home/pi/.config/sense_hat /var/www/.config
```

Now restart the Raspberry Pi. After that, your script should be able to read sensor data over the webserver.

Change Web-Server to run the scripts as user pi (not mandatory)

(instead of www-data):

```
sudo nano /etc/apache2/envvar

Replace www-data by pi (twice)
```

Change Document-Root: /etc/apache2/sites-available/000-default.conf and change DocumentRoot from /var/www/html to whatever you like

Perl example

The following is a perl script that outputs the request headers.

If you have not setup python yet, do that first.

Create /var/www/html/test.pl and enter the following:

```
#!/usr/bin/perl
use strict;
use warnings;
use CGI;

my $q = CGI->new;
my %headers = map {$_ => $q->http($_)} $q->http();

print $q->header(‘text/plain’);
print “Got the following headers:\n”;
for my $header (keys %headers) {
      print “$header: $headers{$header}\n”;
}
```

Open [your-ip-here]/test.pl in a web browser.

Troubleshooting

For anything to do with the webserver, error logs can be found at **/var/log/apache2/error.log**

Complete the following steps if you need to use **sense_hat:**

Execute the following commands as root:

```
# Copy configuration
cp /root/.config/sense_hat_RTIMULib.ini var/www/.config/sense_hat/RTIMULib.ini

# Create udev rules to allow access
touch /etc/udev/rules.d/99-i2c.rules
```

Into the file you created, paste the following:

```
KERNEL==”i2c-[0-7]”,MODE=”0666”
```

```
sudo mkdir /var/www/.config
sudo cp –r /home/pi/.config/sense_hat /var/www/.config
```

Now restart the Raspberry Pi. After that, your script should be able to read sensor data over the webserver.